

Práctica 3 – Cloud Formation

Descripción de los pasos seguidos para cumplir los objetivos


Actividad 1:

1. Describe un Stack de Cloud Formation que incluya los siguientes elementos:
 1. Instancia en EC2 que se pueda acceder por SSH desde el exterior.
 2. Instancia en EC2 que tenga un servidor web en la que muestre alguna característica de la máquina para poder diferenciarla (e.g. nombre, IP, dominio)
 3. Grupos de seguridad y pares de claves (key pairs) que crea necesario


1.) Creamos la instancia SSH

```
SSHGate
1 Resources:
2   SSHGate:
3     Type: 'AWS::EC2::Instance'
4     Properties:
5       InstanceType: t2.micro
6       ImageId: ami-0dbc3d7bc646e8516
7       KeyName: !Ref pra1PEM
8       Tags:
9         - Key: Name
10           Value: SSHGate
11       SecurityGroupIds:
12         - !Ref SecSSH
13
```

2.) Creamos la instancia Servidor Web

```
ServidorWeb   
1 Resources:  
2   ServidorWeb:  
3     Type: 'AWS::EC2::Instance'  
4     Properties:  
5       InstanceType: t2.micro  
6       ImageId: ami-0dbc3d7bc646e8516  
7       KeyName: !Ref pra1PEM  
8       Tags:  
9         - Key: Name  
10          Value: WebServer  
11       SecurityGroupIds:  
12         - !Ref SecWeb  
13       UserData:  
14         'Fn::Base64': !Sub >  
15           #!/bin/bash  
16  
17           yum update -y  
18  
19           yum -y install httpd  
20  
21           systemctl enable httpd  
22  
23           systemctl start httpd  
24  
25           echo '<html><h1>Me llamo Juanjo y me gusta el Tennis</h1></html>' >  
26           /var/www/html/index.html
```

3.) Par de claves(.pem) y Grupos de Seguridad para cada instancia(SSH y Servidor Web)

```
pra1PEM   
1 Resources:  
2   pra1PEM:  
3     Type: 'AWS::EC2::KeyPair'  
4     Properties:  
5       KeyName: pra1PEM  
6
```

```
1 Resources:
2   SecSSH:
3     Type: 'AWS::EC2::SecurityGroup'
4     Properties:
5       GroupDescription: rulesSSH
6       SecurityGroupIngress:
7         - IpProtocol: tcp
8           FromPort: 22
9           ToPort: 22
10          CidrIp: 0.0.0.0/0
11
```

```
1 Resources:
2   SecWeb:
3     Type: 'AWS::EC2::SecurityGroup'
4     Properties:
5       GroupDescription: rulesServer
6       SecurityGroupIngress:
7         - IpProtocol: tcp
8           FromPort: 80
9           ToPort: 80
10          CidrIp: 0.0.0.0/0
11         - IpProtocol: tcp
12           FromPort: 22
13           ToPort: 22
14           SourceSecurityGroupId: !GetAtt SecSSH.GroupId
15
```

En las opciones de la configuración del Stack, seleccionar 'LabRole' como rol.

Permissions

IAM role – optional

Choose the IAM role for CloudFormation to use for all operations performed on the stack.

IAM role name

LabRole

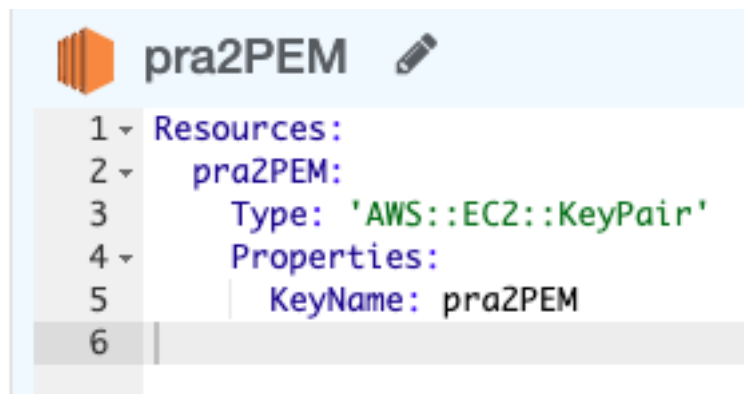
Remove

AWS CloudFormation will use this role for all stack operations. Other users that have permissions to operate on this stack will be able to use this role, even if they don't have permission to pass it. Ensure that this role grants the least privilege.

Actividad 2:

1. Describe un Stack de Cloud Formation que incluya los siguientes elementos:
 1. Dos instancia en EC2 con un servidor web que muestre una pagina similar pero que se pueda reconocer que es un servidor distinto. La pagina tiene que contener alguna característica de la maquina para poder diferenciarla (e.g. nombre, IP, dominio) y ser accesible desde fuera
 2. Un load balancer que distribuya las peticiones entre los dos servidores a partes iguales.
 3. Un “Auto-Scaling Group”(ASG) que tenga como mínimo una instancia y como máximo
2. El ASG debe añadirse al “load balancer” previamente descrito.

- 1.1. Creamos las dos instancias Servidor Web y su Security Group + KeyPair





WebServer1



```
1 ▾ Resources:
2 ▾   WebServer1:
3     Type: 'AWS::EC2::Instance'
4 ▾   Properties:
5     AvailabilityZone: us-east-1c
6     InstanceType: t2.micro
7     ImageId: ami-0dbc3d7bc646e8516
8     KeyName: !Ref pra2PEM
9 ▾   Tags:
10 ▾     - Key: Name
11     |   Value: WebServer1
12 ▾   SecurityGroupIds:
13     - !Ref SecWebServers
14 ▾   UserData:
15 ▾     'Fn::Base64': !Sub >
16     |   #!/bin/bash
17
18     |   yum update -y
19
20     |   yum -y install httpd
21
22     |   systemctl enable httpd
23
24     |   systemctl start httpd
25
26     |   echo '<html><h1>Servidor de Javi 1 </h1></html>' >
27     |   /var/www/html/index.html
28
```



WebServer2



```
1 Resources:
2   WebServer2:
3     Type: 'AWS::EC2::Instance'
4     Properties:
5       AvailabilityZone: us-east-1c
6       InstanceType: t2.micro
7       ImageId: ami-0dbc3d7bc646e8516
8       KeyName: !Ref pra2PEM
9       Tags:
10        - Key: Name
11          Value: WebServer2
12       SecurityGroupIds:
13        - !Ref SecWebServers
14       UserData:
15        'Fn::Base64': !Sub >
16          | #!/bin/bash
17
18          | yum update -y
19
20          | yum -y install httpd
21
22          | systemctl enable httpd
23
24          | systemctl start httpd
25
26          | echo '<html><h1>Servidor de Javi 2 </h1></html>' >
27            /var/www/html/index.html
28
```



SecWebServers



```
1 Resources:
2   SecWebServers:
3     Type: 'AWS::EC2::SecurityGroup'
4     Properties:
5       GroupDescription: SecurityGroup
6       SecurityGroupIngress:
7        - IpProtocol: tcp
8          FromPort: 80
9          ToPort: 80
10         CidrIp: 0.0.0.0/0
11
```

1.2. Creamos el Load Balancer y su Listener + Target Group

LoadBalancer

```
1 ▾ Resources:
2 ▾   LoadBalancer:
3     Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
4 ▾   Properties:
5     Name: LoadBalancer
6     SecurityGroups:
7     - !GetAtt
8       - SecWebServers
9       - GroupId
10 ▾   Subnets:
11     - subnet-0d2f127ba60f42b52
12     - subnet-046b25d65808e65e7
13
```

LBTargGroup

```
1 ▾ Resources:
2 ▾   LBTargGroup:
3     Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
4 ▾   Properties:
5     Name: TargetGroup
6     Protocol: HTTP
7     Port: 80
8     Targets:
9     - Id: !Ref WebServer1
10       Port: 80
11     - Id: !Ref WebServer2
12       Port: 80
13     VpcId: !Ref VPC
14     TargetType: instance
15
```

LBListener

```
1 Resources:
2   LBListener:
3     Type: 'AWS::ElasticLoadBalancingV2::Listener'
4     Properties:
5       DefaultActions:
6         - Type: forward
7           TargetGroupArn: !Ref LBTargGroup
8       LoadBalancerArn: !Ref LoadBalancer
9       Port: 80
10      Protocol: HTTP
11
```

1.3. Creamos el ASG y el Template

WebASG

```
1 Resources:
2   WebASG:
3     Type: 'AWS::AutoScaling::AutoScalingGroup'
4     Properties:
5       AutoScalingGroupName: WebASG
6       AvailabilityZones:
7         - us-east-1c
8         - us-east-1b
9       LaunchTemplate:
10        LaunchTemplateId: !Ref LaunchTemplate
11        Version: !GetAtt LaunchTemplate.LatestVersionNumber
12       TargetGroupARNs:
13         - !Ref LBTargGroup
14       MaxSize: 2
15       MinSize: 1
16
```




LaunchTemplate



```
1 ▾ Resources:
2 ▾   LaunchTemplate:
3     Type: 'AWS::EC2::LaunchTemplate'
4     Properties:
5       LaunchTemplateName: LaunchTemplate
6       LaunchTemplateData:
7         DisableApiTermination: true
8         ImageId: ami-0dbc3d7bc646e8516
9         InstanceType: t2.micro
10        KeyName: !Ref pra2PEM
11 ▾    SecurityGroupIds:
12      - !GetAtt SecWebServers.GroupId
13 ▾    UserData:
14 ▾      'Fn::Base64': !Sub >|
15        |#!/bin/bash
16
17        |yum update -y
18
19        |yum -y install httpd
20
21        |systemctl enable httpd
22
23        |systemctl start httpd
24
25        |echo '<html><h1>Servidor de Javi 999 </h1></html>' >
26        |/var/www/html/index.html
27
```

Creamos un parámetro VPC:

```
1 ▾ Parameters:
2 ▾   VPC:
3     Type: 'AWS::EC2::VPC::Id'
4     Default: vpc-058cda2ce5f98b8ab
5
```

En las opciones de la configuración del Stack, seleccionar 'LabRole' como rol.

Permissions

IAM role – optional

Choose the IAM role for CloudFormation to use for all operations performed on the stack.

IAM role name

LabRole

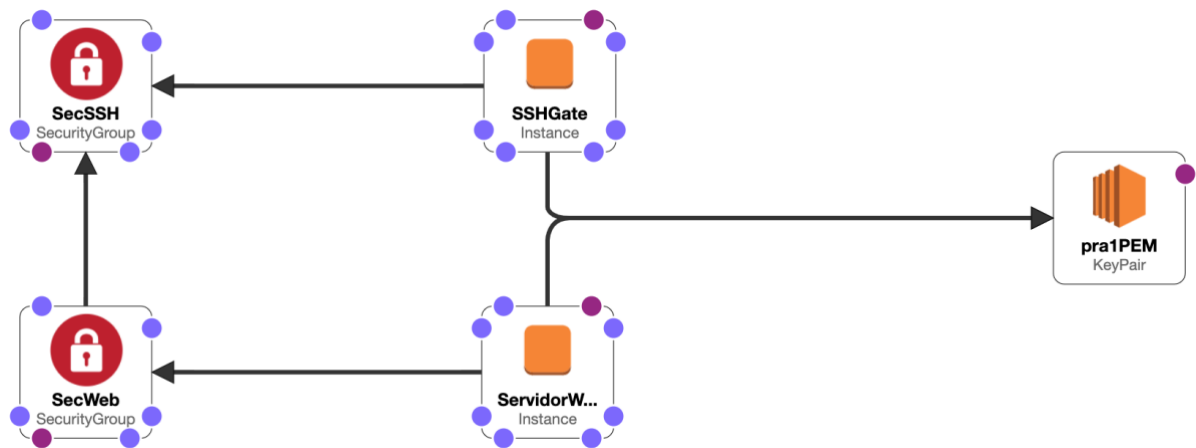
Remove



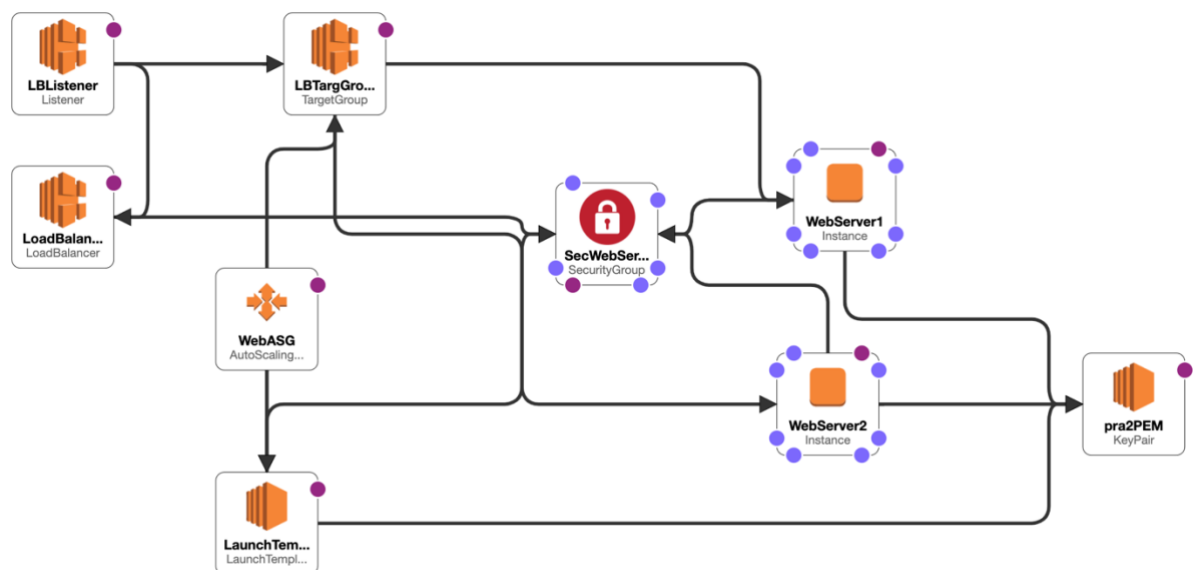
⚠ AWS CloudFormation will use this role for all stack operations. Other users that have permissions to operate on this stack will be able to use this role, even if they don't have permission to pass it. Ensure that this role grants the least privilege.

Diagrama de arquitectura desplegada

Parte 1



Parte 2



Presupuesto y estimación de gasto de los recursos desplegados

